What is claimed is:

1. A system for parsing an arbitrary input stream, comprising:

a plurality of parsers operable to parse an input stream, each parser corresponding to a unique input structure; and

a parser selection agent operable to receive the input stream and select a subset of the plurality of parsers to parse the input stream.

2. The system of Claim 1, wherein the input stream comprises a plurality of differing input structures.

3. The system of Claim 1, wherein the parser selection agent and plurality of parsers are configured in a factory pattern.

4. The system of Claim 1, wherein the parser selection agent is operable to provide to a client, in response to a parse request, at least one of a parser output and an indication when at least some of the input stream is not successfully parsed.

5. The system of Claim 1, wherein the parser selection agent is operable to provide to a client, in response to a parse request, an error message when the parser selection agent is unable to identify one of the plurality of parsers to parse at least some of the input stream and/or when the parser selection agent is able to identify more than one of the plurality of parsers to parse at least some of the input stream.

6. The system of Claim 1, wherein the input stream comprises fault information, the fault information being related to at least one of an alarm and an error, the fault information comprising first fault information related to a first event and in a first format and second fault information related to a second event discrete from the first event and in a second format different from the first format and further comprising:

an encoding agent operable to convert the first and second formats to a common format.

7. The encoding agent of Claim 6, wherein the first and second fault information uses different characters to refer to a same type of event and the encoding agent is further operable to convert the different characters to a common set of characters to refer to the event.

8. A method for parsing an arbitrary input stream, comprising:

(a) receiving an input stream, the input stream comprising information defined by at least first and second input structures;

(b) providing at least a portion of the input stream to at least one of a plurality of parsers;

(c) receiving output from each of the at least one of a plurality of parsers; and

(d) performing at least one of:

(i) selecting a first output from a first parser that corresponds to the first input structure and a second output from a second parser that corresponds to the second input structure; and

(ii) selecting a first parser corresponding to the first input structure to parse one or more first segments of the input stream and a second parser corresponding to the second input structure to parse one or more second segments of the input stream.

9. The method of Claim 8, wherein substep (d)(i) is performed.

10. The method of Claim 8, wherein substep (d)(ii) is performed.

11. The method of Claim 8, wherein each of the plurality of parsers corresponds to a unique set of grammars used by the respective parser when a parse is performed.

12. The method of Claim 8, wherein step (b) comprises:

identifying one or more tokens in the input stream; and

based on the identified one or more tokens, selecting the at least one of a plurality of parsers.

13. The method of Claim 8, wherein step (b) comprises:

determining for each of the at least one of a plurality of parsers whether a match or a no match condition exists.

14. The method of Claim 9, wherein a third parser successfully parses a first portion of the input stream to form a third output and the first parser successfully parses the first portion of the input stream to form a first output and further comprising:

determining which of the first and third outputs most likely corresponds to the first portion.

15. The method of Claim 14, wherein the determining step is performed using a least squares fit analysis.

16. The method of Claim 8, wherein the first parser produces a first output and the first output is a parse tree and further comprising:

recursively evaluating at least some of the nodes in the parse tree to identify nodes requiring additional parsing.

17. The method of Claim 8, wherein the first parser produces a first output and the first output is a parse tree and further comprising:

recursively examining at least some of the nodes in the parse tree to identify nodes of interest to a client.

18. The method of Claim 8, wherein the first parser produces a first output and the first output is a parse tree and wherein at least first and second nodes of the parse tree have differing formats and further comprising:

iteratively traversing a plurality of the nodes of the parse tree to locate nodes of interest, the nodes of interest comprising the first and second nodes; and

converting each of the located nodes of interest to a standard format.

19. The method of Claim 18, wherein each of the first and second nodes use different characters to refer to a same type of event and further comprising:

converting the characters in the first and second nodes to a common set of characters to refer to the type of event.

20. The method of Claim 8, wherein each of the plurality of parsers corresponds to a unique set of tokens and grammar rules.

21. The method of Claim 8, wherein each of the plurality of parsers corresponds to a unique set of attribute grammars.

22. A computer readable medium containing instructions to perform the steps of Claim 8.

23. A method for parsing computer generated information, comprising:

receiving a stream of information, the stream being generated by one of a plurality of possible different computational sources, wherein each computational source generates a stream corresponding to a unique input structure;

comparing at least a portion of the stream with a set of tokens to provide a subset of tokens identified in the at least a portion of the stream;

heuristically identifying, from among at least one of a plurality of possible input structures and a plurality of possible computational sources, at least one of an input structure corresponding to the at least a portion of the stream and a computational source for the at least a portion of the stream; and

parsing the stream based on the identified at least one of an input structure and computational source.

24. The method of Claim 23, wherein each of the tokens has a corresponding method expressing a set of syntactical and/or semantical relationships relating to the respective token and wherein the heuristically identifying step comprises:

for each token in the subset of tokens, invoking a corresponding method.

25. The method of Claim 24, wherein the invoking step comprises

assigning a set of flags a corresponding set of values depending on the presence or absence of a syntactical and/or semantical relationship; and wherein the values of the flags are used to heuristically identify the at least one of an input structure corresponding to the at least a portion of the stream and a computational source for the at least a portion of the stream.

26. The method of Claim 23, wherein a parser performing the steps of Claim 20 is not provided with a flag external to the input stream to identify or assist in the identification of the at least one of an input structure corresponding to the at least a portion of the stream and a computational source for the at least a portion of the stream.

27. A computer readable medium containing instructions to perform the steps of Claim 23.

28. An autonomous heuristic parser, comprising:

an input operable to receive a stream of information, the stream being generated by one of a plurality of possible different computational sources, wherein each computational source generates a stream corresponding to a unique input structure; and

a parser operable to (a) compare at least a portion of the stream with a set of tokens to provide a subset of tokens identified in the at least a portion of the stream; (b) heuristically identify, from among at least one of a plurality of possible input structures and a plurality of possible computational sources, at least one of an input structure corresponding to the at least a portion of the stream and a computational source for the at least a portion of the stream; and (c) parse the stream based on the identified at least one of an input structure and computational source.

29. The parser of Claim 28, wherein each of the tokens has a corresponding parser expressing a set of syntactical and/or semantical relationships relating to the respective token and wherein the parser is further operable, for each token in the subset of tokens, to (d) to invoke a corresponding method.

30. The parser of Claim 29, wherein the parser is further operable to (e) assign a set of flags a corresponding set of values depending on the presence or absence of a syntactical and/or semantical relationship, wherein the values of the flags are used to heuristically identify the at least one of an input structure corresponding to the at least a portion of the stream and a computational source for the at least a portion of the stream.

31. The parser of Claim 28, wherein the parser is not provided with a flag external to the input stream to identify or assist in the identification of the at least one of an input structure corresponding to the at least a portion of the stream and a computational source for the at least a portion of the stream.